GLOBAL OPTIMIZATION OF INTERPLANETARY TRAJECTORIES IN THE PRESENCE OF REALISTIC MISSION CONSTRAINTS

David Hinckley Jr.,‡, Jacob Englander,†, and Darren Hitt‡,

Interplanetary missions are often subject to difficult constraints, like solar phase angle upon arrival at the destination, velocity at arrival, and altitudes for flybys. Preliminary design of such missions is often conducted by solving the unconstrained problem and then filtering away solutions which do not naturally satisfy the constraints. However this can bias the search into non-advantageous regions of the solution space, so it can be better to conduct preliminary design with the full set of constraints imposed. In this work two stochastic global search methods are developed which are well suited to the constrained global interplanetary trajectory optimization problem.

INTRODUCTION

Interplanetary missions are often subject to difficult constraints, like solar phase angle upon arrival at the destination, velocity at arrival, and altitudes for flybys. Preliminary design of such missions is often conducted by solving the unconstrained problem and then filtering away solutions which do not naturally satisfy the constraints. However this can bias the search into non-advantageous regions of the solution space, so it can be better to conduct preliminary design with the full set of constraints imposed. In this work a stochastic global search method is developed which is well suited to the constrained global interplanetary trajectory optimization problem.

In the interplanetary frame, the patched conic approach is used to simplify the planetary arcs. It is often useful in preliminary design to model an interplanetary trajectory using the Multiple Gravity Assist with one Deep-Space Maneuver (MGA-1DSM) problem transcription. This allows for trajectories in which the spacecraft makes a single maneuver between each pair of planetary flybys. In 2007 Vasile, Minisci, and Locatelli used this problem transcription with a robust variant of Differential-Evolution (DE) called Inflationary DE Algorithm. In 2008 Vinkó and Izzo explored solving problems of this type using cooperative combinations of DE, Genetic Algorithm (GA), and Particle Swarm Optimization. They found that combinations of different methods had the potential to surpass component methods on certain problems. This is meaningful to this work as the algorithm presented here draws inspiration from elements of existing evolutionary algorithms. In 2013 Cassioli et al. applied Monotonic Basin Hopping (MBH) and Simulated Annealing to different trajectory design problems. Most recently, Englander solved trajectory optimization problems of the type discussed here using MBH and explicit handling of nonlinear constraints as opposed to a penalty method approach.

‡Mechanical Engineering Graduate Student, University of Vermont
†Aerospace Engineer, Navigation and Mission Design Branch, NASA Goddard Space Flight Center
‡Professor of Mechanical Engineering, University of Vermont
This work is focused on chemical propulsion mission design; as such, the impulsive thrust model is used. The algorithm presented in this work is implemented in the Evolutionary Mission Trajectory Generator (EMTG), NASA Goddard’s open-source interplanetary trajectory optimization tool. EMTG provides all of the necessary modeling for the example problems in this work.

**MGA-1DSM**

The MGA-1DSM problem transcription was first developed by Vasile and De Pascale. Later, Vinkó and Izzo further developed the model. In this model, interplanetary trajectories are described by, after launch, an alternating sequence of impulsive DSMs and unpowered flybys. Each full three-dimensional DSM, the times of flight for the planetary arcs, parameters pertaining to the flybys, as well as launch and arrival condition parameters are all decision variables. Figure 1 illustrates a MGA-1DSM trajectory where \( T_i \) refers to the time of flight for the \( i^{th} \) arc and \( \eta_i \) is the fraction of the arc at which the DSM, shown as a red star, occurs.

![Figure 1. Diagram of a Two-Phase MGA-1DSM Mission](image)

For a more complete explanation of this model see Vinkó and Izzo.

**ALGORITHM: PHASE GENETIC SOLVER**

The algorithm presented in this work is a Phase-Genetic Solver (PGS). Like all evolutionary algorithms, the PGS requires a population of solutions from which to draw genetic material. However unlike other evolutionary algorithms, PGS evaluates only a single candidate solution per
PGS takes advantage of problem-specific gene groupings common to all MGA-1DSM trajectory optimization problems.

PGS maintains a set of rank-sorted archives, or "bins," to provide the required genetic material. The bins are filled with a random assortment of initial genetic material during an initialization period that occurs only once per solver run. Since storing duplicate solutions lessens the total diversity of genetic material stored, a non-exclusionary “kill” distance is employed. If a solution sorted into the archive is within a user-controlled distance of its archive neighbors the less fit (or closest in case the new solution is sufficiently close to both neighbors) is removed. In the MGA-1DSM transcription, groups of decision variables completely describe the planet-to-planet arcs. The fact that when grouped together these decision variables have meaning beyond their independent value was the inspiration for the implementation of a GA-style recombination where these phase gene-blocks are taken from different existing solutions and combined to form a new potential solution.

\[
y = x_1 + F_w (x_2 - x_3)
\]  

(1)

Figure 2 illustrates the phase-wise gene recombination. Since such a recombination cannot create new values for any decision variable, this is paired with a DE-inspired recombination. This operation uses the weighted decision vector combination part of the primary DE creation scheme as an alternative creation method to the GA-style recombination. The switch between the two is a user-controlled parameter set to a value on \([0,1]\) where a uniform random draw falling below the set value triggers solution creation by the DE scheme. This is illustrated by Equation 1.
additional perturbations are added. First, a gene-wise chance of adding a two-tailed Pareto random value was added. Second, a user-controlled likelihood of adding orbital periods to time decision variables was added. These stochastic additions increase the method’s ability to search beyond the limitations of the finite genetic information from which both creation schemes draw.

Step-by-step

The algorithm begins with an initialization period during which trial solutions are drawn from a uniform random distribution over the bounded decision space. The number of trials evaluated during this initialization, i.e. the size of the initial gene pool, is user-controlled but should be set to a number greater than the bin size yet not too close to that number so that to ensure that the bin is filled in the case of duplicates being removed through the “kill” procedure. After the initialization period the main loop begins and continues until one of a set of stopping criteria is met. The main loop begins with creating a new solution by either the GA or DE recombination schemes. Then the element-wise Pareto mutation and the orbit period time hop mutation are applied with the user selected probability. The trial is then locally optimized. For the results shown in this paper local optimization was done using a variation on the Nelder-Mead algorithm that addresses the scale issue in the basins of minima in the MGA-1DSM transcription. The Nelder-Mead local optimizer is run multiple times where each successive run shrinks the initial step distance; this decrease is not dependent on the previous run but the starting position is taken from where the previous run terminated. This is done since very small steps are needed toward the end of the local optimization to narrow in on the minimum, yet starting at those step sizes would greatly increase the time taken for this operation. After local optimization, the solution is sorted into the bin if it is sufficiently fit. This continues until user-specified conditions are met. This work used elapsed time as the primary termination condition. The algorithm is shown in pseudo-code form in Algorithm 1.

TEST PROBLEMS

Testing of the algorithm was conducted of four test problems composed of two types. The first type was European Space Agency Global Trajectory Problem (ESA-GTOP) database inspired problems. The second was derived from real mission objectives and constraints. The purpose of the first set of problems was to provide a means of assessing the method on an understood problem. While the versions of the problems used here do not exactly follow the GTOP specifications, the problems are close enough that answers found should be close to the GTOP posted optimum. The second set of problems demonstrates the ability of the algorithm to address real-world problems that are constrained in meaningful ways. These problems are variations on the OSIRIS-REx mission. For all problems the objective function is minimization of total $\Delta v$.

Problem 1: Cassini 2

The Cassini 2 problem is an academic problem based on a simplified version of the Cassini mission. The version of Cassini 2 here employed closely mirrors the problem as described in the GTOP database. The planetary sequence is identical: launch from Earth with a fly-by sequence of Venus, Venus, Earth, Jupiter ending with a rendezvous maneuver at Saturn. This arrival condition is as specified by the GTOP database. A rendezvous maneuver consumes more fuel than the real Cassini mission’s orbital insertion upon arrival. In this implementation only the total time of flight was bounded as opposed to the bounding of the time-of-flight for planet-to-planet arcs as is the GTOP problem description. The mission’s time of flight was given an upper bound of 8 years. This
Algorithm 1 Phase Genetic Solver

for \( \text{genSw} \) times do
    generate random point \( x \)
    run NLP solver (here Nelder-Mead) to find point \( x^* \) from \( x \)
    if bin is not full or \( F(x^*) < F(x_w) \) where \( x_w \) the worst solution currently in the bin as determined by a weighted sum of objective value and constraint violation then
        Sort(\( x^* \)), Algorithm 2
    end if
    if \( x^* \) is a feasible point then
        save \( x^* \) to archive
    end if
end for

while not hit stop criterion do
    get random value \([0,1]=\text{varType}\)
    if \( \text{varType} < \text{mutCap} = 0.5 \) by default then
        create trial \( x \) through DE-recombination
        \( x = x_1 + 0.85 \cdot (x_2 - x_3) \) for 3 unique \( x_i \)
    else
        create trial \( x \) through GA-recombination
        separate the genome into phase-specific blocks such that:
        \( y = [g_1, g_2, ... g_p] \) where \( g_i \) are vectors of phase-specific gene blocks and \( p \) is the number of phases
        then: \( x = [g_{1v}, g_{2v}, ... g_{pv}] \) where \( g_{iv} \) means take gene-block \( i \) from population member \( v_w \)
    end if
    with probability \( \text{mutUse} = 2\% \) by default, apply Pareto noise to every element of the trial solution
    for each time of flight variable \( t_i \) in \( x' \) do
        if \( \text{rand}(0,1) < \rho_{\text{time-hop}} \) then
            shift \( t_i \) forward or backward one synodic period
        end if
    end for
    run NLP solver to find point \( x^* \) from \( x \)
    if \( F(x^*) < F(x_w) \) then
        Sort(\( x^* \)), Algorithm 2
    end if
    if \( x^* \) is a feasible point then
        save \( x^* \) to archive
    end if
end while
return best \( x^* \) in archive
Algorithm 2 \textit{Sort}(x)

\begin{algorithmic}
\State recursively find the best value beaten
\If {x beats anyone} \then
\State check to see if neighbors are within kill distance
\If {yes} \then
\State kill by proximity then rank, self termination is possible
\EndIf
\If {not self-terminated} \then
\State insert into population
\EndIf
\EndIf
\end{algorithmic}

was selected as the total time bounds as it seemed to be a reasonable amount of time relative to the real mission’s design. The first allowed launch days was set to April 1\textsuperscript{st} 1997.

The spacecraft in the Cassini 2 problem launches on an Atlas V 551 and performs DSMs with a thruster $I_{sp}$ of 320 s. Note that the actual Cassini spacecraft launched on a Titan IVb, but since a Titan IVb performance model was not available Atlas V 551 was used instead. The problem description is summarized in Table 1.

Table 1. Summary of specifications for Problem 1

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Launch date open</td>
<td>4-1-1997</td>
</tr>
<tr>
<td>Launch date close</td>
<td>1-1-2000</td>
</tr>
<tr>
<td>Flight time upper bound</td>
<td>8 years</td>
</tr>
<tr>
<td>Launch vehicle</td>
<td>Atlas V 551</td>
</tr>
<tr>
<td>Spacecraft $I_{sp}$</td>
<td>320</td>
</tr>
<tr>
<td>Planet sequence</td>
<td>E-V-V-E-J-S</td>
</tr>
<tr>
<td>Arrival condition</td>
<td>rendezvous</td>
</tr>
<tr>
<td>Objective function</td>
<td>minimize $\Delta v$</td>
</tr>
<tr>
<td>Run-time bound</td>
<td>2 hours</td>
</tr>
</tbody>
</table>

Problem 2: MESSENGER Reduced

The second test problem is based on the “MESSENGER Reduced” problem from the GTOP database and uses the same flyby sequence. The mission launches from Earth with a fly-by of Earth and two fly-bys of Venus ending with a rendezvous maneuver at Mercury. The total time of flight was bounded to the sum of the bounded legs as given in the GTOP description, 1600 days. This problem was set with the same launch window open date, maximum initial mass, and $I_{sp}$ as Problem 1. The launch vehicle was changed to an Atlas V 401 which is more similar to the Delta II used for the real MESSENGER mission.\textsuperscript{13} This problem’s specifications are summarized in Table 2.

Problem 3 and 4: OSIRIS-REx with Sun-target-angle constraints

Problem 3 and 4 are based on the OSIRIS-REx mission. This mission consists of two journeys, first from Earth to rendezvous with the asteroid Bennu by way of an Earth flyby, then returning to
Earth with an intercept condition. To ensure visibility of the asteroid upon arrival a constraint is placed on the optimization that requires approach within some angular range of the Sun-Bennu line. For Problem 3 the angle is $90^\circ$, and for Problem 4 the angle is $45^\circ$. Since the problem is handed in a patched-conic framework the position at arrival is the center of the asteroid. To account for this the Sun-target-angle is computed using the incoming velocity vector as an approximation to the direction of approach. Figure 3 illustrates the angle defined by this constraint. The yellow circle represents the Sun and the grey circle is Bennu. From this construction it is clear that the angle shown represents a measure of visibility at arrival.

![Figure 3. Sun-Target-Angle](image)

In addition, the arrival velocity at Earth was constrained to be less than $6.28\frac{km}{s}$. This constraint prevents the sample return capsule from burning up on re-entry. Launch and vehicle settings for this problem were set inline with their actual values. The first allowed launch date was September 1st 2016 with a total flight bounds of 10000 days so as to not enforce assumptions on mission time. This is also sensible as the return to Earth is bounded between January 1st 2019 and November 1st 2023. Since this is the only problem where the craft returns from the target, the new decision variable of the stay time at Bennu is introduced. That stay time was given an upper bound of 1500 days. The problem was set to use an Atlas V 411 with an initial mass of 1955kg and an $I_{sp}$ of 230.7s as per the real OSIRIS-REx mission.

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Launch date open</td>
<td>4-1-1997</td>
</tr>
<tr>
<td>Launch date close</td>
<td>1-1-2000</td>
</tr>
<tr>
<td>Flight time upper bound</td>
<td>1600 days</td>
</tr>
<tr>
<td>Launch vehicle</td>
<td>Atlas V 401</td>
</tr>
<tr>
<td>Spacecraft $I_{sp}$</td>
<td>320</td>
</tr>
<tr>
<td>Planet sequence</td>
<td>E-E-V-V-Y</td>
</tr>
<tr>
<td>Arrival condition</td>
<td>rendezvous</td>
</tr>
<tr>
<td>Objective function</td>
<td>minimize $\Delta v$</td>
</tr>
<tr>
<td>Run-time bound</td>
<td>2 hours</td>
</tr>
</tbody>
</table>
RESULTS

Table 4 shows the default settings for PGS as applied to this testing. Deviations from this set will be outlined for specific tests. A setting preceded by “NM” denotes a setting that affects only the Nelder-Mead local optimizer. The settings “stagnation limit” and “repeat” refer to the alterations to the base Nelder-Mead algorithm where “stagnation limit” refers to the number of iterations allowed without improvement and “repeat” refers to the amount of restart and shrinks performed. The “narrowing factor” is the amount by which the Nelder-Mead step size is multiplied upon restart to address the problem of scale inherent in the basins of this problem. Kill distance and initial step here are expressed in the scaled decision space where all decision variables assume values between 0 and 1 which directly map to the entire bounded space. This normalization of the search space is done to improve performance, and has been shown effective with other solvers.²

Problem 1: Cassini 2

The total $\Delta v$ was 8.4878 km/s launching on May 7th 1997. Figure 4 shows the best found solution. The tuning that found this solution did not use the ”kill-distance” duplicate reduction measure. This tuning is more likely to get locally trapped since the GA recombination procedure will increasingly make duplicates once they appear. In 10 trials, the average for this tuning was poor relative to other tunings but the allowed persistence of solutions that are close in the decision space allows for the DE recombination to improve best fitness through more of a local search. It bears mentioning that the amount by which this tuning’s best solution beat tunings with a better average is negligible considering the error inherent in patched conics. For this problem, the best average results were obtained through an increase in the percent likelihood of choosing the DE recombination over the GA recombination. Of the values tested, changing the default 50% trade-off value to a 70% value favoring DE had all of a set of trials tested within 0.04 of the best solution which had a total $\Delta v$ of 8.517 km/s.

These solutions are reasonably close the the GTOP record holding solution whose total $\Delta v$ is 8.383 km/s. Comparing the presented solutions directly with the current record is improper as the problems were not run using the same ephemerides and some details regarding the problem construction are different. The solutions being as close as they are simply shows that the new algorithm presented in this work recovered a sufficiently good solution.

Table 5 shows the various tunings that were explored for this problem.
Table 3. Summary of specifications for Problems 3 & 4

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Journey</strong></td>
<td><strong>Earth to Bennu</strong></td>
</tr>
<tr>
<td>Launch date open</td>
<td>4-1-1997</td>
</tr>
<tr>
<td>Body encounter sequence</td>
<td>E-E-Bennu</td>
</tr>
<tr>
<td>Arrival condition</td>
<td>rendezvous</td>
</tr>
<tr>
<td><strong>Journey</strong></td>
<td><strong>Bennu to Earth</strong></td>
</tr>
<tr>
<td>Wait time lower bound</td>
<td>500 days</td>
</tr>
<tr>
<td>Wait time upper bound</td>
<td>1500 days</td>
</tr>
<tr>
<td>Arrival date bounds</td>
<td>[1-1-2019, 11-1-2023]</td>
</tr>
<tr>
<td>Body encounter sequence</td>
<td>Bennu-E</td>
</tr>
<tr>
<td>Arrival condition</td>
<td>intercept</td>
</tr>
<tr>
<td>Arrival velocity upper bound</td>
<td>$6.28 \frac{km}{s}$</td>
</tr>
<tr>
<td><strong>Global</strong></td>
<td></td>
</tr>
<tr>
<td>Total flight time upper bound</td>
<td>1600 days</td>
</tr>
<tr>
<td>Launch vehicle</td>
<td>Atlas V 401</td>
</tr>
<tr>
<td>Spacecraft $I_{sp}$</td>
<td>320</td>
</tr>
<tr>
<td>Objective function</td>
<td>minimize $\Delta v$</td>
</tr>
<tr>
<td>Total run-time bound</td>
<td>2 hours</td>
</tr>
</tbody>
</table>

Table 4. PGS default settings

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bin size</td>
<td>100</td>
</tr>
<tr>
<td>Number of initialization generations</td>
<td>500</td>
</tr>
<tr>
<td>Enabled proximity kill</td>
<td>true</td>
</tr>
<tr>
<td>Kill distance</td>
<td>0.9</td>
</tr>
<tr>
<td>Fitness penalty weight</td>
<td>100</td>
</tr>
<tr>
<td>DE creation percentage</td>
<td>50</td>
</tr>
<tr>
<td>DE scaling parameter</td>
<td>0.85</td>
</tr>
<tr>
<td>Pareto mutation percentage</td>
<td>2%</td>
</tr>
<tr>
<td>NM max iterations</td>
<td>500</td>
</tr>
<tr>
<td>NM stagnation limit</td>
<td>30</td>
</tr>
<tr>
<td>NM initial step</td>
<td>0.05</td>
</tr>
<tr>
<td>NM repeat</td>
<td>5</td>
</tr>
<tr>
<td>NM narrowing factor</td>
<td>0.01</td>
</tr>
</tbody>
</table>
Figure 4. Cassini 2 result trajectory
<table>
<thead>
<tr>
<th>Test Name</th>
<th>Minimum Fitness</th>
<th>Maximum Fitness</th>
<th>Mean Fitness</th>
<th>Standard Deviation of Fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>NM max iterations = 1000</td>
<td>8.507702203</td>
<td>13.08013145</td>
<td>10.19337892</td>
<td>2.381004889</td>
</tr>
<tr>
<td>NM max iterations = 250 &amp; NM stagnation limit = 30</td>
<td>8.495142971</td>
<td>13.08011605</td>
<td>9.129719809</td>
<td>1.583863494</td>
</tr>
<tr>
<td>Disabled proximity kill</td>
<td>8.487794692</td>
<td>15.4574581</td>
<td>9.517611548</td>
<td>2.019381623</td>
</tr>
<tr>
<td>NM initial step = 0.1</td>
<td>8.519776442</td>
<td>13.80824377</td>
<td>12.40553514</td>
<td>2.155671733</td>
</tr>
<tr>
<td>NM repeat = 3</td>
<td>8.491886681</td>
<td>13.08011288</td>
<td>9.534945467</td>
<td>2.055517186</td>
</tr>
<tr>
<td>NM initial step = 0.01</td>
<td>8.529650524</td>
<td>13.80819269</td>
<td>9.742230666</td>
<td>1.56202128</td>
</tr>
<tr>
<td>Kill distance = 1.25</td>
<td>8.548351326</td>
<td>13.80855295</td>
<td>9.618825493</td>
<td>2.094259801</td>
</tr>
<tr>
<td>NM expansion parameter = 5</td>
<td>8.493211225</td>
<td>13.80810881</td>
<td>9.213243008</td>
<td>1.572387353</td>
</tr>
<tr>
<td>Kill distance = 0.45</td>
<td>8.489449107</td>
<td>15.64071313</td>
<td>10.84552859</td>
<td>2.550076981</td>
</tr>
<tr>
<td>Kill distance = 1.5</td>
<td>8.548151998</td>
<td>8.6373963</td>
<td>8.588254452</td>
<td>0.028344059</td>
</tr>
<tr>
<td>DE creation percentage = 60</td>
<td>8.510255988</td>
<td>13.80812768</td>
<td>10.37414168</td>
<td>2.37686051</td>
</tr>
<tr>
<td>NM max iterations = 250</td>
<td>8.502856018</td>
<td>13.80823687</td>
<td>10.28526561</td>
<td>2.330492865</td>
</tr>
<tr>
<td>Fitness penalty weight = 50</td>
<td>8.510825618</td>
<td>9.445452757</td>
<td>8.696852204</td>
<td>0.354515305</td>
</tr>
<tr>
<td>DE creation percentage = 30</td>
<td>8.943567626</td>
<td>16.04331072</td>
<td>13.08227751</td>
<td>2.115409215</td>
</tr>
<tr>
<td>DE creation percentage = 80</td>
<td>8.518332019</td>
<td>8.57894109</td>
<td>8.54124301</td>
<td>0.015297718</td>
</tr>
<tr>
<td>NM stagnation limit = 50</td>
<td>8.500648814</td>
<td>13.80810493</td>
<td>9.037603714</td>
<td>1.590176354</td>
</tr>
<tr>
<td>Fitness penalty weight = 10</td>
<td>8.50422969</td>
<td>44.25559316</td>
<td>17.57454411</td>
<td>11.51069538</td>
</tr>
<tr>
<td>Default</td>
<td>8.501509161</td>
<td>15.64214462</td>
<td>11.20484013</td>
<td>2.95955532</td>
</tr>
<tr>
<td>DE creation percentage = 70</td>
<td>8.517078919</td>
<td>8.556319048</td>
<td>8.534003909</td>
<td>0.009798648</td>
</tr>
<tr>
<td>Kill distance = 2.0</td>
<td>8.638286021</td>
<td>13.87964676</td>
<td>10.2763394</td>
<td>2.346441835</td>
</tr>
<tr>
<td>Pareto mutation percentage = 100</td>
<td>8.501850121</td>
<td>11.16184864</td>
<td>8.972233049</td>
<td>0.816102361</td>
</tr>
</tbody>
</table>

Table 5. Tunings tested for Problem 1
Problem 2: MESSENGER Reduced

For this problem the total $\Delta v$ was 7.848 km/s launching on April 25th 1998. The delivered mass was 326 kg with a flight time of 4.19 years. Figure 5 shows the trajectory. This solution was found by altering the Nelder-Mead local optimizer settings to increase the maximum number of iterations allowed and increasing the number of iterations without improvement before returning a value. This was not the best solution found in terms of $\Delta v$. Solutions to this academically posed problem were found with lower total $\Delta v$ but they, due to how this problem is posed, perform maneuvers that are not applicable in real mission flight. Figure 6 shows the best solution found to this problem. The total $\Delta v$ was 7.700 km/s which is considerably better than the previous solution. The launch date was May 12th 1998, the delivered mass was 360 kg, and the total flight time was 4.36 years. The problem with this solution is that it essentially performs most of the rendezvous as part of a deep-space maneuver and then follows Mercury through most of an orbit. This is due to the rendezvous condition which is not a realistic arrival condition. Real missions would either perform an intercept, for direct landing, or an orbital insertion upon arrival. Neither of these conditions allow for this matching of Mercury for most of an orbit to be locally advantageous. In this case, the academic formulation of the MESSENGER-Reduced problem is not an accurate representation of the true preliminary design problem. However it is used here for the sake of comparison to other published work.

Figure 5. MESSENGER Reduced result trajectory
For this problem, although comparison is improper, because our bounds on the flight times between bodies are slightly different than those found in the GTOP version of MESSENGER-reduced, the best solution found beats the GTOP record whose $\Delta v$ is 8.630 km/s. As was the case with Problem 1, the only conclusion that can be drawn from this comparison is that the new optimizer presented in this work is capable of addressing this class of problem given some consideration to tuning.

Table 6 shows the tunings tested for this problem. Different versions of the MESSENGER problem were tested before testing the version shown here. Since the other problem variations had more success with this set of tunings than with others, including default, only these were tested.
<table>
<thead>
<tr>
<th>Test Name</th>
<th>Minimum Fitness</th>
<th>Maximum Fitness</th>
<th>Mean Fitness</th>
<th>Standard Deviation of Fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE creation percentage = 30</td>
<td>7.700553833</td>
<td>10.077949</td>
<td>8.69836204</td>
<td>0.957901076</td>
</tr>
<tr>
<td>NM max iterations = 1000 &amp; NM stagnation limit = 100</td>
<td>7.885601368</td>
<td>9.838366048</td>
<td>8.135282451</td>
<td>0.580954014</td>
</tr>
<tr>
<td>Pareto mutation percentage = 100</td>
<td>7.848294059</td>
<td>9.196548589</td>
<td>8.092378624</td>
<td>0.424963406</td>
</tr>
<tr>
<td>NM max iterations = 100</td>
<td>7.883441903</td>
<td>11.46388641</td>
<td>9.370414423</td>
<td>1.286407838</td>
</tr>
<tr>
<td>Disabled proximity kill</td>
<td>7.885493493</td>
<td>11.46414506</td>
<td>8.830137907</td>
<td>1.18791056</td>
</tr>
<tr>
<td>NM max iterations = 100</td>
<td>7.884693973</td>
<td>10.5763399</td>
<td>8.818799601</td>
<td>0.85652511</td>
</tr>
<tr>
<td>NM repeat = 3</td>
<td>7.891120342</td>
<td>9.968710382</td>
<td>8.988314296</td>
<td>0.931023514</td>
</tr>
<tr>
<td>DE creation percentage = 70</td>
<td>7.884667061</td>
<td>9.950609605</td>
<td>8.665714852</td>
<td>0.95438583</td>
</tr>
<tr>
<td>NM stagnation limit = 50</td>
<td>7.884242668</td>
<td>9.951119188</td>
<td>8.163303062</td>
<td>0.632826718</td>
</tr>
<tr>
<td>NM stagnation limit = 100</td>
<td>7.883914168</td>
<td>11.46634384</td>
<td>9.96979681</td>
<td>1.206551117</td>
</tr>
<tr>
<td>NM initial step = 0.01</td>
<td>7.883914168</td>
<td>11.46634384</td>
<td>9.96979681</td>
<td>1.206551117</td>
</tr>
</tbody>
</table>

Table 6. Tunings tested for Problem 2
Problem 3: OSIRIS-REx: 90° Sun-Target-Angle

The total $\Delta v$ for the best solution found was 0.7238 km/s launching on September 23rd 2016. The $\Delta v$ for the journey to Bennu was 0.4624 km/s arriving on December 22nd 2018. The delivered mass to Bennu was 1594 kg. The departure date from Bennu was May 14th 2021. The return date to Earth was September 25th 2023 with a final mass of 1420 kg. Figure 7 shows the trajectory of this solution.

The tuning that produced this result, like that of Problem 1, was the disabling of the duplicate control measure. This led to premature convergence in many of the trials but also allowed for both recombination types to act locally. For this problem the local solver, whose default settings suited the Problem 1, required more adjustment than the algorithm. The best average was obtained through increasing the number of iterations of the Nelder-Mead algorithm and increasing the number of number of generations without improvement allowed before the solution was considered locally optimal. The best found solution of this setting was only 0.0004 km/s worse than the result shown here.

Table 7 lists the tunings tested. Initial testing on an unconstrained version of this problem revealed much of the deviation required from default tuning was in the local optimizer. This makes sense as the default tunings for both the method and local optimization are based on testing for Problem 1. As of yet, a tuning law for all problems has not been established.
<table>
<thead>
<tr>
<th>Test Name</th>
<th>Minimum Fitness</th>
<th>Maximum Fitness</th>
<th>Mean Fitness</th>
<th>Standard Deviation of Fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disabled proximity kill</td>
<td>0.723849797</td>
<td>2.932235497</td>
<td>1.173036182</td>
<td>0.68831895</td>
</tr>
<tr>
<td>NM initial step = 0.1</td>
<td>0.725279214</td>
<td>1.796980154</td>
<td>0.835174524</td>
<td>0.320672698</td>
</tr>
<tr>
<td>NM stagnation limit = 50</td>
<td>0.724487126</td>
<td>1.233349354</td>
<td>0.810434537</td>
<td>0.154609675</td>
</tr>
<tr>
<td>NM max iterations = 1000</td>
<td>0.724595703</td>
<td>1.608428143</td>
<td>0.954556275</td>
<td>0.291337706</td>
</tr>
<tr>
<td>NM stagnation limit = 100</td>
<td>0.724276918</td>
<td>1.607709109</td>
<td>0.852585628</td>
<td>0.260491888</td>
</tr>
<tr>
<td>Kill distance = 1.2</td>
<td>0.725769679</td>
<td>1.288751624</td>
<td>0.802798213</td>
<td>0.16791975</td>
</tr>
<tr>
<td>Kill distance = 0.45</td>
<td>0.724017532</td>
<td>1.607853071</td>
<td>0.898080663</td>
<td>0.258426999</td>
</tr>
<tr>
<td>DE creation percentage = 30</td>
<td>0.72518409</td>
<td>17.29093509</td>
<td>3.433378663</td>
<td>4.746951776</td>
</tr>
<tr>
<td>NM max iterations = 1000 &amp; NM stagnation limit = 100</td>
<td>0.724384127</td>
<td>873.0522079</td>
<td>122.7224839</td>
<td>257.1555035</td>
</tr>
<tr>
<td>DE creation percentage = 70</td>
<td>0.725050606</td>
<td>1.609754007</td>
<td>0.869859487</td>
<td>0.256750254</td>
</tr>
<tr>
<td>Pareto mutation percentage = 100</td>
<td>0.724466525</td>
<td>3.738515419</td>
<td>1.107804668</td>
<td>0.881684002</td>
</tr>
<tr>
<td>Default</td>
<td>0.725657145</td>
<td>3.738515419</td>
<td>1.107804668</td>
<td>0.881684002</td>
</tr>
</tbody>
</table>

Table 7. Tunings tested for Problem 3
Problem 4: OSIRIS-REx: 45° Sun-Target-Angle

For this more constrained version of the previous problem the best total $\Delta v$ found was 0.7241 km/s launching on the same day as the previously reported result. The $\Delta v$ for the journey to Bennu was 0.4625 km/s arriving on the same day as the solution presented for Problem 3. The delivered mass to Bennu was also the same. The departure date from Bennu was May 15th 2021. The return date to Earth and the final mass were also the same as the result for Problem 3. The solution to Problem 4 here is just an adjustment of Problem 3’s solution that achieves the new Sun-target-angle constraint.

The tuning that produced this result was also the tuning that best solved the problem in the most trials. The only adjustment relative to the default tuning was increasing the Nelder-Mead local solver’s number of allowed generations without improvement.

Table 8 shows the tunings tested for Problem 4. These tests are identical to those for Problem 3 since the difference between the two problems is the adjustment of a single constraint. While this problem is harder due to this different constraint it was not thought too different to affect the general pattern of tunings explored.
<table>
<thead>
<tr>
<th>Test Name</th>
<th>Minimum Fitness</th>
<th>Maximum Fitness</th>
<th>Mean Fitness</th>
<th>Standard Deviation of Fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disabled proximity kill</td>
<td>0.794681232</td>
<td>3.797895363</td>
<td>1.64365942</td>
<td>1.101750074</td>
</tr>
<tr>
<td>Pareto mutation percentage = 100</td>
<td>0.799880971</td>
<td>6.16061388</td>
<td>1.468606482</td>
<td>1.583540108</td>
</tr>
<tr>
<td>NM max iterations = 1000</td>
<td>0.724436189</td>
<td>4.045370978</td>
<td>1.159974362</td>
<td>0.987205563</td>
</tr>
<tr>
<td>NM stagnation limit = 100</td>
<td>0.724134498</td>
<td>1.068555671</td>
<td>0.801919046</td>
<td>0.124221252</td>
</tr>
<tr>
<td>Default</td>
<td>0.762818765</td>
<td>1.64313778</td>
<td>0.957427469</td>
<td>0.248590476</td>
</tr>
<tr>
<td>Kill distance = 1.2</td>
<td>0.800867037</td>
<td>1.83435561</td>
<td>0.943784705</td>
<td>0.309631387</td>
</tr>
<tr>
<td>Kill distance = 0.45</td>
<td>0.754246276</td>
<td>5.67378172</td>
<td>1.451568898</td>
<td>1.443897955</td>
</tr>
<tr>
<td>NM initial step = 0.1</td>
<td>0.797778251</td>
<td>2.36178032</td>
<td>0.993787507</td>
<td>0.461541481</td>
</tr>
<tr>
<td>DE creation percentage = 70</td>
<td>0.798337374</td>
<td>1.084989391</td>
<td>0.879957468</td>
<td>0.105577135</td>
</tr>
<tr>
<td>NM max iterations = 1000 &amp; NM stagnation limit = 100</td>
<td>0.795939001</td>
<td>0.96675381</td>
<td>0.836301535</td>
<td>0.065568903</td>
</tr>
<tr>
<td>DE creation percentage = 30</td>
<td>0.810500627</td>
<td>13.98726773</td>
<td>2.472446933</td>
<td>3.860479861</td>
</tr>
<tr>
<td>NM stagnation limit = 50</td>
<td>0.80221576</td>
<td>2.052751678</td>
<td>0.966982707</td>
<td>0.368824685</td>
</tr>
</tbody>
</table>

Table 8. Tunings tested for Problem 4
CONCLUSION

In summary, a new global stochastic optimizer is presented that is shown capable of solving interplanetary trajectory optimization problems well within reasonable time-frames. The varied tunings tested for these problems are given so as to show the need for proper tuning for a given problem. While no universal tuning law exists at the moment, problem length in terms of journey and flyby sequence is currently thought to be the biggest factor in how the algorithm must be tuned. This is beneficial as the problem’s length is known before optimization, thus tunings that hinge upon this can be settled without a trial and error approach. Many of the tunings explored in this work were focused on refining the local optimizer which is wholly separate from the actual algorithm.

Future Work

As of this writing the algorithm has only been tested when coupled with a Nelder-Mead penalty-based local optimizer but the method is independent of local optimizer choice and can easily interface with others, such as the Sparse Nonlinear OPTimizer (SNOPT). Knowledge of the behavior of the method as generations progress is based on the assumption that unfeasible solutions are allowed to be returned with a penalty definition of fitness. The effect of discarding infeasible solutions may reduce the genetic diversity of the bin in a deleterious manner but the extent of such potential effects is unknown.

This algorithms approach of GA-style recombination is not limited to MGA-1DSM problems. This method can be applied to any optimization problem where disjoint groupings of decision variables have distinct meaning. An appropriate renaming of the optimizer for such an extension would the Codon Genetic Solver, borrowing terminology from biology.

The most important work to be done is to determine a tuning law that is applicable to a wide range of problems. This would allow PGS to be useful in the context of a hybrid optimal control tool, where the individual trajectory optimization problems are generated in real-time by a discrete “outer-loop” solver. In such an environment there is no opportunity for a human analyst to study each individual trajectory design problem and assign appropriate tunings. Future work on this project will be focused on determining such a generic tuning law.

ACKNOWLEDGMENT

This work was partially supported by the Vermont Space Grant under NASA Cooperative Agreement # NNX10AK674.

REFERENCES


